ᴜ

UNITED STATES PATENT AND TRADEMARK OFFICE

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/828,573	04/21/2004	Matthew A. Ahrens	03226.390001; P8399	5289

32615   7590   05/23/2007

OSHA LIANG L.L.P./SUN
1221 MCKINNEY, SUITE 2800
HOUSTON, TX 77010

EXAMINER
DEBNATH, SUMAN

ART UNIT	PAPER NUMBER
2135	

MAIL DATE	DELIVERY MODE
05/23/2007	PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

	Application No.	Applicant(s)
Office Action Summary	10/828,573	AHRENS ET AL.
	Examiner	**Art Unit**	
	Suman Debnath	2135	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE _3_ MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1) ☒ Responsive to communication(s) filed on _21 April 2004_.

2a) ☐ This action is **FINAL**.   2b) ☒ This action is non-final.

3) ☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4) ☒ Claim(s) _1-26_ is/are pending in the application.

     4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5) ☐ Claim(s) _____ is/are allowed.

6) ☒ Claim(s) _1-26_ is/are rejected.

7) ☐ Claim(s) _____ is/are objected to.

8) ☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9) ☐ The specification is objected to by the Examiner.

10) ☐ The drawing(s) filed on _____ is/are: a) ☐ accepted or b) ☐ objected to by the Examiner.

     Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

     Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11) ☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12) ☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

     a) ☐ All   b) ☐ Some * c) ☐ None of:

        1. ☐ Certified copies of the priority documents have been received.

        2. ☐ Certified copies of the priority documents have been received in Application No. _____.

        3. ☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

     * See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)

2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)

3) ☒ Information Disclosure Statement(s) (PTO/SB/08)
Paper No(s)/Mail Date _03/20/2007_.

4) ☐ Interview Summary (PTO-413)
Paper No(s)/Mail Date. _____.

5) ☐ Notice of Informal Patent Application

6) ☐ Other: _____.

DETAILED ACTION

1.   Claims 1-26 are pending in this application.

2.   Examiner has pointed out particular references contained in the prior arts of record in the body of this action for the convenience of the applicant. Although the ·specified citations are representative of the teachings in the art and are applied to the specific limitations within the individual claim, other passages and figures may apply as well. Applicant should consider the entire prior art as applicable as to the limitations of the claims. It is respectfully requested from the applicant, in preparing the response, to consider fully the entire references as potentially teaching all or part of the claimed invention, as well as the context of the passage as taught by the prior arts or disclosed by the examiner.

Claim Rejections - 35 USC § 102

3.   The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

A person shall be entitled to a patent unless –

(b) the invention was patented or described in a printed publication in this or a foreign country or in public use or on sale in this country, more than one year prior to the date of application for patent in the United States.

4.   Claims 1-26 are rejected under 35 U.S.C. 102(e) as being anticipated by Talagala et al. (Pub. No.: US 2002/0161972 A1), hereinafter "Talagala".

5.   As to claim 1, Talagala discloses a method for storing a data block (abstract),

comprising: storing the data block in a storage pool ("...the host machine interacts with

the storage array via virtual address" and explains that "when a block is written, a

physical location is chosen for it" e.g., see –[0059]); obtaining a data block location

([0017], [0059]); calculating a data block checksum for the data block ([0017], [0058] –

[0059]); and storing a first indirect block in the storage pool, wherein the first indirect

block comprises the data block location and the data block checksum (FIG. 6C, [0059],

"An indirection map (e.g. block remapping table) matches virtual block addresses to

physical block addresses. Block-level checksums may be provided in the indirection

map").

6.   As to claim 2, Talagala discloses calculating a first indirect block checksum

([0017], [0058] – [0059]); obtaining a first indirect block location (FIG. 6C, [0059]); and

storing a second indirect block in the storage pool, wherein the second indirect block

comprises the first indirect block location and the first indirect block checksum (Talagala

teaches writing a block having a checksum and a location in PGT (Parity Group Table)

which have a "segment" field to store a location and a "checksum" field to store a

checksum and provides which contain parity/stripe group tables having entries for

multiple blocks of data stored in a storage pool, e.g. see – [0059] and FIG. 6C, 7B and

8B).

7.   As to claim 3, Talagala discloses the method further comprising: assembling the

first indirect block (Talagala discloses this concept as "a has indirect table (HIT)" which

"maps virtual block addresses to an entry or index number in a parity group table" – e.g.

see, [0054] – [0055] and FIG. 6B, 7A, 8A, 6C, 7B and 8B. Note that each virtual address

in "HIT" maps to an entry in "PGT (parity group table)" and that each field in PGT table

stores configuration information for each "indirect block")

8.   As to clam 4, Talagala discloses wherein assembling the first indirect block

comprises: storing the data block checksum in a checksum field in a block pointer in the

first indirect block (Talagala teaches this concept as "each block's checksum is stored in

an entry in the indirection map, e.g. as part of a block remapping table entry (e.g. PGT

entry) for each block" –e.g. see, [0058] and FIG. 6C, 7B and 8B, wherein each entry

representing/pointing to a block in PGT contains a checksum in a checksum field), and

storing the data block location in the block pointer, wherein storing the data block

location comprises storing a metaslab ID (Talagala teaches this concept as "HIT (Hash

Indirection 'Table)" which contains ".PGT (Parity Group Table)" indices to access a PGT

which contains configuration information for each of the parity groups/metaslabs such

as a segment field which indicated the physical disk location (disk and segment) of

parity groups, e.g. see – [0055], [0058], [0059] and FIG. 6B, 6C, 7A, 7B, 8A and 8B.

Talagala provides an example in which Virtual address zero corresponds to PGT index

12, which contains valid data at physical segment D1.132 wherein "this may be

interpreted as Disk 1, segment 132", e.g. see – [0057] and FIG. 6B, 6C, 7A, 7B, 8A and

8B) and offset (Talagala teaches this concept as entries stored under the "Next Entry In

Parity Group" field in PGT (parity group table) which points to the next virtual block

entry, e.g. see – [0057] and FIG. 6B, 6C, 7A, 7B, 8A and 8B).

9.   As to claim 5, Talagala discloses storing a birth value in a birth field in the block

pointer (FIG. 8A and 8B, [0065], Talagala teaches this concept as an embodiment

having "a hashed indirection table (HIT) which maintains generational images" and

explains that "the PGT index columns are now labeled version zero through version two,

where version zero corresponds to the most current version and version two

corresponds to the oldest version").

10.   As to claim 6, Talagala discloses wherein the first indirect block is assembled

using a data management unit (FIG. 2 and 3, [0033], "Storage Controller 401").

11.   As to claim 7, Talagala discloses wherein the storage pool comprises at least

one storage device (FIG. 2 and 3, [0033], "array of storage devices 410").

12.   As to claim 8, Talagala discloses wherein the storage pool is divided into a

plurality of metaslabs (Talagala teaches having different "stripes of data" within an array

storage devices – e.g. see FIG. 4, [0043] and explains that a "stripe" of data is

analogous of a "parity group" –e.g. see [0063], wherein configuration information for

each of the "parity groups" is stored in a "PGT (parity group table)" –e.g. see FIG. 6C,

7B and 8C, which are equivalent to metaslabs as claimed).

13.   As to claim 9, Talagala discloses wherein each of the plurality of metaslabs is

associated with a metaslab ID (Talagala discloses this concept as each virtual block has

a virtual address which is used to access a "HIT (Hash Indirection Table)" which

contains "PGT (Parity Group Table) indices to access a PGT which contains

configuration information for each of the parity groups/metaslabs such as a segment

field which indicated the physical disk location (disk and segment) of parity groups. e.g.

see- FIG. 6B, 6C, 7A, 7B, 8A and 8B, [0055], [0058], [0059]).

14.   As to claim 10, Talagala discloses wherein the data block location comprises the

metaslab ID and an offset (Talagala teaches this concept as each virtual block has a

virtual address which is used to access a "HIT (Hash Indirection Table)" which contains

"PGT (Parity Group Table)" indices to access a PGT which contains configuration

information for each of the parity groups/metaslabs. Talagala also provides an example

in which Virtual address zero corresponds to PGT index 12, which contains valid data at

physical segment D1.132 wherein "this may be interpreted as Disk 1, segment 132",

e.g. see – [0057] and FIG. 6B, 6C, 7A, 7B, 8A and 8B. Talagala also teaches "next

entry in Parity Group" field in PGT (Parity Group Table) which points to the next virtual

block entry as an offset, e.g. see – [0057] and FIG. 6B, 6C, 7A, 7B, 8A and 8).

15.   As to claim 11, Talagala discloses wherein storing the data block comprises

using a storage pool allocator (FIG. 2 and 3, [0033], "Storage Controller 401").

16.   As to claim 12, Talagala discloses a method for storing a first data block and a

second data block (abstract), comprising: storing the first data block and the second

data block in a storage pool ([0015], [0017], [0058] – [0059]); obtaining a first data block

location and a second data block location ([0016], [0017], [0058] – [0059]); calculating a

first data block checksum for the first data block ("...a plurality of data blocks and a

parity block calculated for the data blocks" e.g., see- [0016], [0017], [0058] – [0059]);

calculating a second data block checksum for the second data block ([0016], [0017],

[0058] – [0059], [0060]); and storing an array of block pointers in an indirect block

([0013], [0052]-[0053], [0060], [0065] and FIG. 6A, 6B, 6C), wherein the array block of

pointers comprises, a first block pointer comprising the first data block location and the

first data block checksum ([0013], [0052]-[0053], [0060] , [0065] and FIG. 6A, 6B, 6C),

and a second block pointer comprising the second data block location and the second

data block checksum ([0013], [0052]-[0053], [0060]).

17.   As to claim 13, Talagala discloses wherein the indirect block is assembled using

a data management unit (FIG. 2 and 3, [0033], "Storage Controller 401").

18.   As to claim 14, Talagala disclose wherein the indirect block is stored using a

storage pool allocator (FIG. 2 and 3, [0033], "Storage Controller 401").

19.   As to claim 15, Talagala discloses a method for retrieving data in a data block

(abstract), comprising: obtaining an indirect block comprising a stored checksum and a

data block location (Talagala teaches this concept as "HIT (Hash Indirection 'Table)"

which contains "PGT (Parity Group Table)" indices to access a PGT which contains

configuration information for each of the parity groups/metaslabs such as a segment

field which indicated the physical disk location (disk and segment) of parity groups, e.g.

see – [0055], [0058], [0059] and FIG. 6B, 6C, 7A, 7B, 8A and 8B.); obtaining the data

block using the data block location ([0061], "When a block read is requested, the block's

indirection map entry is read to find the block's physical address"); calculating the

checksum for the data block to obtain a calculated checksum (Talagala teaches this

concept as "each block's checksum is stored in an entry in the indirection map, e.g. as

part of a block remapping table entry (e.g. PGT entry) for each block" –e.g. see, [0058]

and FIG. 6C, 7B and 8B, wherein each entry representing/pointing to a block in PGT

contains a checksum in a checksum field); retrieving the data from the data block, if the

stored checksum equals the calculated checksum ("The block received from the READ

may be compared to the checksum received from the READ " –e.g., [0040], [0059], FIG.

2, 3, 6C, 7B and 8B); and performing an appropriate action, if the stored checksum is

not equal to the calculated checksum ("The block received from the READ may be

compared to the checksum received from the READ (e.g. by recalculating the

checksum from the data and comparing the new checksum to the read checksum). If

they do not match, an error may be detected immediately" —e.g., see, [0040], [0059],

FIG. 2, 3, 6C, 7B and 8B).

20.   As to claim 16, Talagala discloses wherein the calculated checksum is calculated

using a storage pool allocator (FIG. 2, 3, 6C and [0033], [0059], "Storage Controller

401").

21.   As to claim 17, Talagala discloses a method for storing and retrieving a data

block (abstract), comprising: storing the data block ([0017], [0059]); obtaining a data

block location ([0017], [0059]); calculating a data block checksum for the data block

(Talagala teaches this concept as "each block's checksum is stored in an entry in the

indirection map, e.g. as part of a block remapping table entry (e.g. PGT entry) for each

block" —e.g. see, [0058] and FIG. 6C, 7B and 8B, wherein each entry

representing/pointing to a block in PGT contains a checksum in a checksum field);

storing a block pointer in an indirect block, wherein the block pointer comprises the data

block location and the data block checksum (FIG. 8A and 8B, [0065], Talagala teaches

this concept as an embodiment having "a hashed indirection table (HIT) which

maintains generational images" and explains that "the PGT index columns are now

labeled version zero through version two, where version zero corresponds to the most

current version and version two corresponds to the oldest version"); obtaining the

indirect block comprising the block pointer ([0013], [0052]-[0053], [0060], [0065] and

FIG. 6A, 6B, 6C); obtaining the data block using the data block location stored in the

block pointer ([0013], [0052]-[0053], [0060], [0065] and FIG. 6A, 6B, 6C); calculating the

checksum for the data block to obtain a calculated checksum ([0016], [0017], [0058] –

[0059]); retrieving data from the data block, if the data block checksum stored in the

block pointer equals the calculated checksum ([0040], [0059], FIG. 2, 3, 6C, 7B and 8B);

and performing an appropriate action, if the data block checksum is not equal to the

calculated checksum ([0040], [0059], FIG. 2, 3, 6C, 7B and 8B).

22.   As to claim 18, Talagala discloses a system for storing a data block, comprising:

a storage pool comprising the data block and a first indirect block, wherein the first

indirect block comprises a data block checksum and a data block location (Talagala

teaches this limitation as "An indirection map (e.g. block remapping table) matches

virtual block addresses to physical block addresses. Block-level checksums may be

provided in the indirection map", e.g., see – [0059] and FIG. 6C); and a storage pool

allocator configured to store the data block and the first indirect block in the storage pool

(FIG. 2 and 3, [0033], "Storage Controller 401").

23.   As to claim 19, Talagala discloses the system further comprising: a second

indirect block, comprising a first indirect data block checksum and a first indirect block

location, wherein the storage pool allocator is further configured to store the second

indirect block in the storage pool (Talagala teaches writing a block having a checksum

and a location in PGT (Parity Group Table) which have a "segment" field to store a

location and a "checksum" field to store a checksum and provides which contain

parity/stripe group tables having entries for multiple blocks of data stored in a storage

pool, e.g. see – [0059] and FIG. 6C, 7B and 8B).

24.   As to claim 20, Talagala discloses further comprising: a data management unit

configured to assemble the first indirect block and request the storage pool allocator to

store the first indirect block (FIG. 2 and 3, [0033], "Storage Controller 401").

25.   As to claim 21, Talagala discloses wherein the storage pool comprises at least

one storage device (FIG. 2 and 3, [0033], "array of storage devices 410").

26.   As to claim 22, Talagala discloses wherein the storage pool is divided into a

plurality of metaslabs (Talagala teaches having different "stripes of data" within an array

storage devices – e.g. see FIG. 4, [0043] and explains that a "stripe" of data is

analogous of a "parity group" –e.g. see [0063], wherein configuration information for

each of the "parity groups" is stored in a "PGT (parity group table)" –e.g. see FIG. 6C,

7B and 8C, which are equivalent to metaslabs as claimed).

27.   As to claim 23, Talagala discloses wherein each of the plurality of metaslabs is

associated with a metaslab ID (Talagala discloses this concept as each virtual block has

a virtual address which is used to access a "HIT (Hash Indirection Table)" which

contains "PGT (Parity Group Table) indices to access a PGT which contains

configuration information for each of the parity groups/stripes/metaslabs such as a

segment fiend which indicated the physical disk location (disk and segment) of parity

groups. e.g. see- FIG. 6B, 6C, 7A, 7B, 8A and 8B, [0055], [0058], [0059]).

28.   As to claim 24,Talagala discloses wherein the data block location comprises the

metaslab ID and an offset (Talagala teaches this concept as each virtual block has a

virtual address which is used to access a "HIT (Hash Indirection Table)" which contains

"PGT (Parity Group Table)" indices to access a PGT which contains configuration

information for each of the parity groups/metaslabs. Talagala also provides an example

in which Virtual address zero corresponds to PGT index 12, which contains valid data at

physical segment D1.132 wherein "this may be interpreted as Disk 1, segment 132",

e.g. see – [0057] and FIG. 6B, 6C, 7A, 7B, 8A and 8B. Talagala also teaches "next

entry in Parity Group" field in PGT (Parity Group Table) which points to the next virtual

block entry as an offset, e.g. see – [0057] and FIG. 6B, 6C, 7A, 7B, 8A and 8).

29.   As to claim 25, Talagala discloses a computer system for storing a data block,

comprising: a processor (FIG. 2, item 100); a memory (FIG. 2, item 200); a storage

device (FIG. 2, item 400); and software instructions stored in the memory for enabling

the computer system under control of the processor, to: store the data block in a

storage pool ([0033] and [0054], "To facilitate keeping track of the data and parity

information, a block remapping technique may be implemented in software and/or

hardware which maps a logical or virtual block address to a physical storage device

segment"); obtain a data block location ([0017], [0059], see also [0051]); calculate a

data block checksum for the data block ([0059] and [0051]); and store a first indirect

block in the storage pool, wherein the first indirect block comprises the data block

location and the data block checksum (FIG. 6C, [0059], "An indirection map (e.g. block

remapping table) matches virtual block addresses to physical block addresses. Block-

level checksums may be provided in the indirection map").

30. As to claim 26, Talagala discloses a network system having a plurality of nodes,

comprising: a storage pool comprising the data block and a first indirect block ([0017],

[0059], [0061]), wherein the first indirect block comprises a data block checksum and a

data block location (FIG. 6C, [0059]); and a storage pool allocator configured to store

the data block and the first indirect block in the storage pool (FIG. 2 and 3, [0033],

"Storage Controller 401"), wherein the storage pool is located on any one of the plurality

of nodes ([0017], [0059], [0051]), and wherein the storage pool allocator is located on

any one of the plurality of nodes ([0017], [0033], [0059], [0051]) and FIG. 6C).

Conclusion

31. The prior art made of record and not relied upon is considered pertinent to

applicant's disclosure. See accompanying PTO 892.

- US 2005/0091229 A1 – Verification of file system log data using per-entry
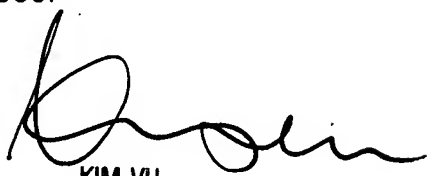 checksums.

- US 7,181,585 – Defensive heap memory management.

32.   Any inquiry concerning this communication or earlier communications from the

examiner should be directed to Suman Debnath whose telephone number is 571 270

1256. The examiner can normally be reached on 8 am to 5 pm.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's

supervisor, Kim Y. Vu can be reached on 571 272-3859. The fax phone number for the

organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the

Patent Application Information Retrieval (PAIR) system. Status information for

published applications may be obtained from either Private PAIR or Public PAIR.

Status information for unpublished applications is available through Private PAIR only.

For more information about the PAIR system, see http://pair-direct.uspto.gov. Should

you have questions on access to the Private PAIR system, contact the Electronic

Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a

USPTO Customer Service Representative or access to the automated information

system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

SD
SD

KIM VU
SUPERVISORY PATENT EXAMINER
TECHNOLOGY CENTER 2100